

Die Aufgaben müssen von jedem Teilnehmer einzeln bearbeitet und abgegeben werden. Die Abgabe muss **bis spätestens Montag 22.06.2014 um 23:59 Uhr** über unser Hausaufgabenverwaltungssystem <https://se.cs.uni-kassel.de/hms/> erfolgen (nicht per Email). Die Abgabe ist nur als einzelne *.zip Datei möglich. Daher müssen alle für eine Abgabe relevanten Daten zu einem solchen Archiv kombiniert werden.

Hinweise zur Abgabe:

- Die Hausaufgabe ist in Form von **einem** exportierten Eclipse Projekten abzugeben. Mit Hilfe der Eclipse Export Funktion ist es möglich mehrere Projekte **in einer zip-Datei** zu exportieren. Ist ein Projekt nicht korrekt exportiert, kann es bei der Korrektur nicht berücksichtigt werden (es bietet sich also an, den Import des exportierten Projektes auszuprobieren).

WICHTIG: Benennen Sie ihre Projekte für diese Abgabe nach folgendem Schema:

DPSS15_HA<a>__<Matrikelnummer>,

wobei <a> für die aktuelle Hausaufgabe und für die Aufgabennummer steht.

Beispiel für Aufgabe 1:

DPSS15_HA3_1_12345678.

- Libraries müssen in einen lib oder im Projekt eingefügt werden oder als weitere Projekt mit dem Namen DPSS15_HA<a>_libs_<Matrikelnummer>. Es darf keine Abhängigkeit zu anderen Projekten bestehen oder zu externen Libraries.

Allgemeines:

Orientiert euch für die Lösung der Aufgaben an den zugehörigen Übungen und Vorlesungen:
<http://seblog.cs.uni-kassel.de/category/ss15/designpatternss15/>

Zusammensetzung der Note:

Es wird n Hausaufgaben geben, von denen **n-2 Sinnvoll** bearbeitet werden müssen.

Sinnvoll bedeutet mindestens 50% der Punkte.

Am Ende ergibt sich die Note aus dem Durchschnitt n-2 besten Abgaben.

Zusätzlich wird es zwei Zusatzaufgaben geben mit denen fehlenden Abgaben ausgeglichen werden können.

Die Zusatzaufgaben werden deutlich umfangreicher als normale Hausaufgaben sein und dienen nicht zur Verbesserung der Note.

Zusatzaufgaben müssen mit **80%** bestanden werden, um wie folgt im Durchschnitt der n-2 besten Hausaufgaben berücksichtigt zu werden:

Wenn bei einer Zusatzaufgabe **100%** der Punkte erreicht wurden, wird sie wie eine **40%** Hausaufgabe bei der Berechnung berücksichtigt. Bei **90%** wird sie wie eine **30%** Aufgabe bewertet, bei **80%** wird sie wie eine **20%** Aufgabe bewertet.

Die Veranstaltung gilt als nicht bestanden, wenn mehr als **zwei Hausaufgaben nicht abgegeben** wurden oder der Mittelwert der zur Benotung herangezogenen Abgaben **weniger als 50%** beträgt.

Fragen bitte an hahn@uni-kassel.de

Aufgabe 1 Plugin Pattern

- In dieser Abgabe soll ein Pizza Konfigurator mit GUI gebaut werden :)
- Jede Zutat soll als Plugin für den PizzaKonfigurator realisiert werden.
 - Es sollen mindestens 5 verschiedene Zutaten realisiert werden
 - Jedes Plugin ist als eigenes Projekt zu Realisieren
 - Benennt die Projekte wie folgt:
DPSS15_HA9_1_<Matrikelnummer>_<pluginname>
 - Vom PizzaKonfiguratorProjekt aus darf es keine Abhängigkeiten zu den Plugins geben
 - Die Plugins sollen ein Button mit Bild in die Pizza Konfigurator GUI einfügen. Nach Auswahl einer Zutat, soll es möglich sein, diese der Pizza hinzuzufügen. Dabei soll darauf Einfluss genommen werden können, ob die Zutat aus einer anderen zuvor ausgewählten Zutat liegt oder daneben
- Der Konfigurator soll eine Möglichkeit haben neue Plugins zur Laufzeit hinzuzufügen. Es muss also einen Button geben, der einen FileChooser oder Textfeld öffnet, wo eine neue Plugin-Jar angegeben werden kann, die dann geladen wird.
- Der Konfigurator soll mit einem Argument von der Kommandozeile gestartet werden. Das Argument soll auf ein Verzeichnis verweisen, in dem zum Start zuladende Plugin-Jars liegen. Diese Plugins sollen dann auch beim Start geladen werden.
 - Bsp: `java -jar PizzaKonfigurator.jar ./myPizzaPlugins/`
- Zwischen den einzelnen Zutaten soll es eine „liegtAuf“ und „liegtNeben“ Beziehung geben
- Die fertige Pizza soll im Konfigurator graphisch angezeigt werden und als JSON File exportiert werden können
- Haltet die GUI so einfach wie möglich

Beispiel JSON Export:

```
{ name: „SuperPizza“,  
  topping: [ { name: „Tomatensauce“,  
             topping:[  
               { name: „Salami“, topping:[ { name: „Käse“ } ] },  
               { name: „Salami“, topping:[ { name: „Käse“ } ] },  
               { name: „Salami“, topping:[ { name: „Käse“ } ] }  
             ]  
          } ]  
}
```